

A Polynomial Time Approximation Scheme for a Single Machine Scheduling Problem Using a Hybrid Evolutionary Algorithm

Boris Mitavskiy* and Jun He

Department of Computer Science

Aberystwyth University, Aberystwyth, SY23 3DB, U.K.

February 9, 2012

Abstract

Nowadays hybrid evolutionary algorithms, i.e. heuristic search algorithms combining several mutation operators some of which are meant to implement stochastically a well known technique designed for the specific problem in question while some others playing the role of random search, have become rather popular for tackling various NP-hard optimization problems. While empirical studies demonstrate that hybrid evolutionary algorithms are frequently successful at finding solutions having fitness sufficiently close to the optimal, many fewer articles address the computational complexity in a mathematically rigorous fashion. This paper is devoted to a mathematically motivated design and analysis of a parameterized family of evolutionary algorithms which provides a polynomial time approximation scheme for one of the well-known NP-hard combinatorial optimization problems, namely the “single machine scheduling problem without precedence constraints”. The authors hope that the techniques and ideas developed in this article may be applied in many other situations.

1 Introduction

Scheduling problems appear naturally in a variety of applications whenever one needs to make decisions about the order in which tasks (such as parallel subroutines in computer programs, transportation of passengers or goods, arranging examination timetables in universities etc.) are to be executed economically subject to limited resources (such as limited number of processors, vehicles, employees, etc.) and other constraints (such as the timing when various tasks become available or dependence of a certain task upon the outcome of another task etc.) As it is usually the case in combinatorial optimization, most scheduling problems are NP-hard (see, for instance [7] and [6]) and, therefore, various heuristic search algorithms are frequently exploited to tackle them. Among these are the so-called hybrid evolutionary algorithms. Such techniques combine random mutation or recombination operators with other algorithms (such as various greedy approaches, for instance). A hybrid evolutionary algorithm for Graph Coloring developed in [9] embeds local search into the framework of evolutionary algorithms. This hybrid evolutionary algorithm combines a new class of highly specialized crossover operators and a well-known tabu search algorithm. Experiments with such a hybrid algorithm have been carried out on large DIMACS Challenge benchmark graphs and the results have proven very competitive with and even better than those of state-of-the-art algorithms. A Hybrid evolutionary algorithm for Job Scheduling presented in [4] combines memetic algorithms and several local search algorithms. The memetic algorithm is used as the principal heuristic that guides the search and can use any of the 16 local search algorithms during the search process. The local search algorithms used in combination with the MA have been obtained by fixing either the type of the neighborhood or the type of the move; these include swap/move based search, Hill Climbing, Variable Neighborhood Search, and Tabu Search. A popular approach to design hybrid evolutionary algorithms is to combine local search with evolutionary algorithms.

*Corresponding author.

Application-specific, parameterized local search algorithms (PLSAs), in which optimization accuracy can be traded off with run time, arise naturally in many optimization contexts. In [8], a novel approach, called simulated heating, for systematically integrating parameterized local search into evolutionary algorithms (EAs) has been introduced. Using the framework of simulated heating, both static and dynamic strategies for systematically managing the tradeoff between PLSA accuracy and optimization effort have been investigated. The goal was to achieve maximum solution quality within a fixed optimization time budget. It has been shown that the simulated heating technique better utilizes the given optimization time resources than standard hybrid methods that employ fixed parameters, and that the technique is less sensitive to these parameter settings. This framework has been applied to three different optimization problems, and the results have been compared to the standard hybrid methods, showing quantitatively that careful management of this tradeoff is necessary to achieve the full potential of an EA/PLSA combination.

Despite their frequent success the choice of a particular algorithm to optimize a certain problem is mainly driven by the intuition or personal bias of its designer towards a certain technique rather than by solid mathematical reasoning so that the reasons behind the high performance usually remain unclear (see [10] as well as [3]).

In the current paper we design a parameterized family of hybrid 2+2 EAs based on rigorous mathematical approach motivated by the theory presented in [7], that provide a polynomial time approximation scheme for a “single machine scheduling problem”. This type of scheduling problem will be described in detail in section 3, nonetheless it is important to know that the problem has been shown to be strongly NP-hard in [6] via a polynomial time reduction from the classical 3-PARTITION problem. Our hybrid 2 + 2-EA combines a local mutation operator which is based on a greedy heuristic called the Jackson rule together with a given approximation ratio parameter $\epsilon > 0$ and a global mutation operator applied with relatively small probability with the aim to explore all the possible local optima. At the end of section 3 we prove (see theorem 16) that given any specified approximation parameter $\epsilon > 0$, our 2 + 2-EA finds a schedule within the $(1 + \epsilon)$ factor from an optimal schedule after $n^{\frac{1}{\epsilon}+7} + n^5$ time steps with overwhelmingly high probability: at least $1 - \left(\frac{1}{k \cdot n}\right)^n$ for all sufficiently large n , where k is a constant while n is the total number of jobs in the schedule. As a matter of fact, thanks to parts 1 and 5 of theorem 2, the constant k can be made arbitrarily close to 1 as long as it is strictly smaller than 1. As a compromise, making k closer to 1 increases the lower bound on the total number of jobs n beyond which the complexity bound holds.

2 Asymptotic Notation and Elementary Mathematical Tools Involved in the Design and Complexity Analysis

The following notation is frequently used in algorithm complexity theory.

Definition 1. Given sequences of real numbers $f(n)$ and $g(n)$ (i.e. functions f and $g : \mathbb{N} \rightarrow \mathbb{R}$) we say that $f = O(g)$ if $\exists k > 0$ and $N > 0$ such that $\forall n > N$ we have $f(n) \leq k \cdot g(n)$. Dually, we say that $f = \Omega(g)$ if $\exists k > 0$ and $N > 0$ such that $\forall n > N$ we have $f(n) \geq k \cdot g(n)$.

Recall from elementary analysis that the irrational number $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \approx 2.71$ up to two significant digits exists and is commonly denoted by the letter e (see, for instance, [11]). The elementary facts summarized in theorem 2 below will be frequently exploited throughout the next section.

Theorem 2. *The following facts are true:*

$$\forall x \in \mathbb{R} \text{ we have } \lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n = e^x. \quad (1)$$

In particular, when $x = -1$ we have

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = e^{-1}. \quad (2)$$

Let $\beta > \alpha > 0$ and consider the sequence

$$s_n = \left(1 - \frac{1}{n^\alpha}\right)^{n^\beta} = \left(\left(1 - \frac{1}{n^\alpha}\right)^{n^\alpha}\right)^{n^{\beta-\alpha}}. \quad (3)$$

Then $\forall k < 1 \exists N \in \mathbb{N}$ such that $\forall n > N$ we have

$$s_n < e^{-k \cdot n^{\beta-\alpha}}. \quad (4)$$

Dually, assume that $\alpha > \beta > 0$. Then $\forall k < 1 \exists N \in \mathbb{N}$ such that $\forall n > N$ we have

$$s_n > e^{-k \cdot n^{\beta-\alpha}}. \quad (5)$$

$$\forall x > 0 \text{ the inequality } e^{-x} > 1 - x \text{ holds.} \quad (6)$$

Proof. 1 and 2 can be found in any introductory real analysis textbook: see, for instance, [11]. In summary, the arguments are based on binomial expansion and power series. To see inequalities 4 and 5, observe that the function $\varphi(x) = (1 - \frac{1}{x})^x$ is strictly decreasing on the interval $(0, \infty)$ since $\varphi'(x) < 0$. Now let $k < 1$ be given. Monotonicity together with the fact in 2 tell us that $\lim_{x \rightarrow \infty} (1 - \frac{1}{x})^x = e^{-1}$ so that $\exists N \in \mathbb{N}$ large enough such that $\forall n > N$ we have

$$e^{-k} > \left(\left(1 - \frac{1}{n^\alpha} \right)^{n^\alpha} \right) > e^{-1}$$

so that the inequalities 4 and 5 follow at once. The inequality 6 is very commonly used in practice and has a very short proof: consider the function $f(x) = e^{-x} - (1 - x)$. Then $f(1) = 0$ and $\forall x > 0$ we have $f'(x) > 0$ so that f is strictly increasing on the interval $(0, \infty)$. In particular, $\forall x \in (0, \infty)$ we have $f(x) > f(1) = 0$ which is equivalent to the desired inequality in 6. \square

Apart from the elementary notions summarized above, basic probability theory and basic Markov chain theory are heavily exploited in the upcoming section. There is a plenty of literature covering both subjects in much detail: see, for instance, [5].

3 A Parameterized Family of Hybrid 2+2EAs that Provides a Polynomial Time Approximation Scheme for Single-Machine Scheduling Problem

The ideas in this section are closely related to these presented in [7]. Suppose we have a sequence of n jobs $\{J_i\}_{i=1}^n$ where each job J_i must be processed without interruption for a time $p_i > 0$ on the same machine M . A job J_i is released at time $r_i \geq 0$ associated with it and it becomes available for processing only at the time r_i and any time after as long as the machine M is not occupied at the time being. As soon as a job J_i has finished processing it is sent for delivery immediately. A specific job J_i has its own delivery time q_i . Here we assume that there is no restriction on the total number of jobs being delivered simultaneously. Our objective is to find a reordering of the sequence $\{J_i\}_{i=1}^n$ of jobs which minimizes the minimal time when all of the jobs have just been delivered, referred to as the maximal lateness of the schedule.¹ In summary, given a set of n ordered triplets $\{J_i\}_{i=1}^n$ with each $J_i = (r_i, p_i, q_i)$ that stand for *release time*, *processing time* and *delivery time* of the job J_i respectively, our aim is the following: given $\epsilon > 0$, produce a $\mu + \mu$ hybrid EA which produces a schedule (i.e. a feasible² reordering/permutation π_t of the sequence $\{J_i\}_{i=1}^n$ in time t depending polynomially on n and $\frac{1}{\epsilon}$ which, in turn, determines the new sequence $(J_{\pi_t(i)})_{i=1}^n$) the maximal lateness of which, call it L_t , is such that

$$L_t \leq (1 + \epsilon) \cdot L^* \quad (7)$$

where L^* denotes the optimal (i.e. the minimal possibly achievable) maximal lateness. We will say that such a schedule L_t is ϵ -optimal. Let $P = \sum_{i=1}^n p_i$ and $\delta = \epsilon \cdot P \leq \epsilon \cdot L^* \leq \epsilon \cdot L^* + \delta$ so that we can write $(1 + \epsilon) \cdot L^* \geq L^* + \delta$.

¹It has been explicitly shown in [7] that the setting above is equivalent to the model with due dates in place of the delivery times via a simple linear change of variables, yet the model with delivery times is a lot better suitable for the algorithm design and analysis.

²In the special case of single machine scheduling with no preprocessing constraints studied in the current article, as we will see, every permutation of the sequence $\{J_i\}_{i=1}^n$ of jobs produces a feasible schedule.

Thereby to deduce inequality 7 it is sufficient to establish the inequality 8 below which is expressed purely in terms of δ and n :

$$L_t \leq L^* + \delta. \quad (8)$$

This form is more convenient for the algorithm description and analysis. We now proceed to describe the search space in detail.

The search space consists of all possible permutations of the sequence $\{J_i\}_{i=1}^n$ of jobs. We will denote the search space with

$$\Omega = \{\pi \mid \pi \text{ is a permutation on } \{1, 2, \dots, n\}\}. \quad (9)$$

where $\pi \in \Omega$ represents the schedule $(J_{\pi(i)})_{i=1}^n$.

For theoretical convenience we introduce an extra job $J_0 = (0, 0, 0)$ into the schedule that will make no practical difference at all.

Definition 3. Extend the initial sequence $\{J_i\}_{i=1}^n$ of jobs to include the job J_0 having the property that $r_0 = p_0 = q_0 = 0$ (i.e. the new initial schedule is now $\{J_i\}_{i=0}^n$ where $J_0 = (0, 0, 0)$). We extend every permutation π on $\{i \mid 1 \leq i \leq n\}$ to the permutation π on $\{i \mid 0 \leq i \leq n\}$ trivially by letting $\pi(0) = 0$ (in fact, this is obviously the unique extension).

Our hybrid 2 + 2 EA will ensure that we deliver the ϵ -optimal schedule within time polynomial in n depending on δ . In order to construct and to analyze (as well as to motivate the construction of) the 2 + 2 EA it is convenient to introduce the following recursive definitions:

Definition 4. We will write $s_i(\pi)$ to denote the *starting time* of the job $J_{\pi(i)}$ within the schedule π . Let $s_0(\pi) = 0$. For $i \geq 1$ let $s_i(\pi) = \max\{s_{i-1}(\pi) + p_{\pi(i-1)}, r_{\pi(i)}\}$. Let $L_\pi = \max_{1 \leq i \leq n} s_i(\pi) + p_{\pi(i)} + q_{\pi(i)}$ denote the *maximal lateness* (i.e. the earliest time when all the jobs have just been delivered) of the schedule π .

When the schedule π is clear from the context we will write s_i in place of $s_i(\pi)$.

Definition 4 makes perfect sense since the jobs can be started only after they have been released and there is no point to wait for a job to start unless the machine is occupied. It is worth pointing out the following.

Remark 5. $\forall i$ with $1 \leq i \leq k$ there are three possible mutually exclusive cases.

Case 1: $s_{i-1} + p_{\pi(i-1)} < r_{\pi(i)}$. In this case

$$s_i = r_{\pi(i)} > s_{i-1} + p_{\pi(i-1)}.$$

Case 2: $s_{i-1} + p_{\pi(i-1)} > r_{\pi(i)}$. In this case

$$s_i = s_{i-1} + p_{\pi(i-1)} > r_{\pi(i)}.$$

Case 3: $s_{i-1} + p_{\pi(i-1)} = r_{\pi(i)}$. In this case

$$s_i = s_{i-1} + p_{\pi(i-1)} = r_{\pi(i)}.$$

Definition 6. Given $\delta > 0$, let $A_\delta = \{i \mid p_i < \delta\}$ and $B_\delta = \{i \mid p_i \geq \delta\}$ denote the indices of these jobs in the initial sequence that have processing times shorter than δ and at least as long as δ respectively. Given a subset of indices $I = \{i_1, i_2, \dots, i_b\} \subseteq \{1, 2, 3, \dots, n\}$ with $|I| = |B_\delta| = b$ and a bijection $\phi : I \rightarrow B_\delta$ we will say that a schedule (a permutation determining the schedule) π is (k, δ, ϕ) -Jackson if $\forall j$ with $1 \leq j \leq k$ the following are true:

If $j = i_q \in I$ then $\pi(j) = \phi(i_q)$.

If $j \notin I$ then exactly one of the three cases above takes place with j playing the role of i .

If case 1 takes place then we require that $\forall l > i$ we have $r_{\pi(l)} \geq r_{\pi(j)}$ and, in case if $r_{\pi(j)} = r_{\pi(l)}$ we must have $q_{\pi(l)} \leq q_{\pi(j)}$.

If cases 2 or 3 take place then we require that $\forall l > j$ we have either $r_{\pi(l)} > s_j(\pi)$ or $(r_{\pi(l)} \leq s_j(\pi) \text{ and } q_{\pi(l)} \leq q_{\pi(j)})$.

An (n, δ, ϕ) -Jackson schedule will be called simply a (δ, ϕ) -Jackson schedule.

Remark 7. Notice that given any schedule $\pi \exists! \phi$ such that π is a (k, δ, ϕ) -Jackson schedule. Indeed, ϕ is obtained by identifying the locations of the jobs in B_δ within the schedule π . Setting $k = 0$ will always do, but one can agree to select the maximal k such that π is a (k, δ, ϕ) -Jackson schedule and such k is evidently unique. From now on whenever we refer to (k, δ, ϕ) -Jackson schedule we will have in mind that k is maximal with respect to this property.

In view of remark 7 we have an important equivalence relation on the search space of all the schedules Ω (see 9).

Definition 8. We say that $\pi \sim \sigma$ if both, π and σ , are (k_1, δ, ϕ) -Jackson and (k_2, δ, ϕ) -Jackson respectively with the common unique ϕ . Let

$$\mathcal{X}(\phi) = \{\pi \mid \pi \text{ is } (k, \delta, \phi)\text{-Jackson where } 0 \leq k \leq n\}$$

denote the equivalence class of schedules corresponding to the given ϕ .

A crucial fact behind the design and success of our algorithm is the following clever and elegant theorem which is proved without being explicitly stated in section 1.2.3, chapter 1 of [7].

Theorem 9. *Given any initial sequence $\{J_i\}_{i=1}^n$ of jobs with each $J_i = (r_i, p_i, q_i)$, $\forall \delta > 0 \exists$ an indexing set $I \subseteq \{1, 2, \dots, n\}$ with $I = |B_\delta|$ and a bijection $\phi : I \rightarrow B_\delta$ such that any (δ, ϕ) -Jackson schedule is ϵ -optimal.*

Our 2+2-EA always keeps the fittest individual as the second individual in the population. This individual is never mutated, while the first individual is mutated at every time step. The mutant always replaces the first individual and if the mutant is fitter than the second individual it replaces the second individual as well. There will be two mutation operators involved in the algorithm: one local, driving towards achieving (δ, ϕ) -Jackson schedule, and another global: searching for a specific ϕ which is guaranteed to exist in theorem 9. The local mutation operator is applied with relatively large probability while the global mutation with small probability so as to ensure that the algorithm has time to find a (δ, ϕ) -Jackson schedule before it explores a different choice of the bijection ϕ . Notice that there are $\prod_{i=0}^{|B_\delta|-1} (n-i) < n^{|B_\delta|}$ of such bijections while

$$|B_\delta| \cdot \delta \leq \sum_{i \in B_\delta} p_i \leq \sum_{i=1}^n p_i = P$$

so that $|B_\delta| \leq \frac{P}{\delta} = \frac{1}{\epsilon}$ (see inequalities 7 and 8 and discussions preceding these inequalities) and this immediately implies the following important lemma which appears implicitly in section 1.2.3 of [7].

Lemma 10. *The total number of ways to position the jobs from the set B_δ within a schedule, (in other words, the total number of ways to select a bijection ϕ) is bounded above by $n^{\frac{1}{\epsilon}}$.*

Lemma 10 tells us that the “quotient search space” on which the global mutations act is polynomial in size and this fact will be important for the design and runtime complexity analysis of our 2+2-EA. Another important fact contributing to the design and complexity analysis of our algorithm is the following.

Proposition 11. *Suppose the initial sequence of jobs $\{J_i\}_{i=1}^n$ has the property that all the jobs have pairwise distinct delivery times: $\forall i$ and j with $1 \leq i < j \leq n$ we have $q_i \neq q_j$. Choose any $k \geq 1$ and a bijection $\phi : I \rightarrow B_\delta$. Let π and σ denote any two (k, δ, ϕ) -Jackson schedules. Then $\forall i$ with $1 \leq i \leq k$ we have $s_i(\pi) = s_i(\sigma)$ and $\pi(i) = \sigma(i)$. In particular, a (δ, ϕ) -Jackson schedule is unique.*

Proof. We argue by the least natural number principle. Suppose proposition 11 is not true. In this case \exists a minimal $j \in \{i \mid 1 \leq i \leq k\}$ such that either one of the assertion of proposition 11 fails. Definition 3 tells us that $\pi(0) = \sigma(0) = 0$ so that $j > 0$ trivially. Observe also that according to definition 6 $j \notin I$. Consider now the jobs $J_{\pi(j)}$ and $J_{\sigma(j)}$ where $j \notin I \cup \{0\}$. By minimality of $j \forall i$ with $0 \leq i \leq j$ we have $\pi(i) = \sigma(i)$, $r_{\pi(i)} = r_{\sigma(i)}$, $p_{\pi(i)} = p_{\sigma(i)}$ and $s_i(\pi) = s_i(\sigma)$. First we observe that $r_{\pi(j)} \neq r_{\sigma(j)}$ (otherwise by definition 6 we must have $q_{\pi(j)} = q_{\sigma(j)}$ contrary to the assumption). Next we note that one of the three cases described in remark 5 must take place for π .

case 1: In this case, notice that the subcase $r_{\pi(j)} < r_{\sigma(j)}$ is impossible by definition 6. The subcase $r_{\sigma(j)} < r_{\pi(j)}$ is dual to the previous subcase when exchanging the roles of π and σ .

cases 2 or 3: Without loss of generality assume that case 2 or case 3 takes place for the schedule σ as well (otherwise apply the argument in case 1 swapping the roles of π and σ). Now that $r_{\pi(j)} \leq s_j$ and $r_{\sigma(j)} \leq s_j$, by definition 6 we must have $q_{\pi(j)} = q_{\sigma(j)}$ contrary to the assumption once again.

Thus the minimal $j \in \{i \mid 1 \leq i \leq k\}$ such that at least one of the assertions in proposition 11 fails has the property that $\pi(j) = \sigma(j)$. The minimality of j together with definition 4 immediately entail the remaining property in proposition 11 that $s_i(\pi) = s_i(\sigma)$ thereby contradicting the defining property of j and the desired conclusion now follows by the least natural number principle. \square

In view of proposition 11 it is convenient to introduce the following definition:

Definition 12. We say that an instance of a scheduling problem $\{J_i\}_{i=1}^n$ is *generic* if $\forall i$ and j with $1 \leq i < j \leq n$ we have $q_i \neq q_j$.

While the condition of being “generic” ensures the conclusion of proposition 11, it is hardly a restriction for two reasons. The first reason is that tiny perturbations of delivery times will cause a very small change in the maximum lateness of the schedule and that this change can be made arbitrarily small depending on the size of the perturbation. The second reason is that one can simply introduce additional total orders on the equivalence classes of jobs having equal delivery times. Now if there is an ambiguity whether to schedule a job J_i or J_j in a specific position, it must be the case that $q_i = q_j$ (as we have seen from the proof of proposition 11) so that the additional total order on the equivalence class $[J_i] = [J_j]$ will determine the job to be scheduled uniquely. Apparently any permutation of generic schedule remains generic. In view of the comments above we will assume from now on that the instance of our scheduling problem is a generic schedule.

We introduce two mutation operators, one “local” and one “global”.

Local Mutation Operator $\mu_\delta^{\text{local}}$: Given a schedule π select indices i and j where $1 \leq \pi(i) < \pi(j) \leq n$ uniformly at random.

IF $p_{\pi(i)} < \delta$ and $p_{\pi(j)} < \delta$ (i.e. if $\pi(j)$ and $\pi(i) \in A_\delta$ which is the same as saying that $\pi(i) \notin I$ and $\pi(j) \notin I$) **THEN** there are two cases

Case 1: $r_{\pi(i)} > s_i$. In this case there are further three subcases:

Subcase 1: $r_{\pi(i)} = r_{\pi(j)}$. In this subcase, **IF** $q_{\pi(j)} > q_{\pi(i)}$ **THEN** set $\pi := \pi \circ (i, j)$ (i.e. swap jobs in position i and j within the schedule π) **ELSE** set $\pi := \pi$ (i.e. do nothing).

Subcase 2: $r_{\pi(i)} > r_{\pi(j)}$. In this subcase set $\pi := \pi \circ (i, j)$ (i.e. perform the swap of jobs in positions i and j within the schedule π).

Subcase 3: $r_{\pi(i)} < r_{\pi(j)}$. In this subcase set $\pi := \pi$ (i.e. do nothing)

Case 2: $r_{\pi(i)} \leq s_i$. In this case **IF** $r_{\pi(j)} \leq s_i$ and $q_{\pi(j)} > q_{\pi(i)}$ **THEN** set $\pi := \pi \circ (i, j)$ **ELSE** set $\pi := \pi$ (i.e. do nothing).

ELSE set $\pi := \pi$ (i.e. do nothing).

Essentially, the local mutation is designed to modify the schedule to become (δ, ϕ) -Jackson step by step as described precisely below.

Lemma 13. Let π be a generic schedule (see definition 12 and the comments that follow) and let k be the maximal integer such that π is (k, δ, ϕ) -Jackson. (Notice that such k always exists since every schedule is $(0, \delta, \phi)$ -Jackson.) After an application of the local mutation operator $\mu_\delta^{\text{local}}$ if $k < n$ then with probability bigger than $\frac{1}{n^2}$ the schedule π is updated to become $(k+1, \delta, \phi)$ -Jackson and otherwise k remains the maximal integer such that π is (k, δ, ϕ) -Jackson. If, on the other hand, the schedule π is already (δ, ϕ) -Jackson (i.e. $k = n$) then it remains such with probability 1.

Proof. By proposition 11 the jobs in positions 1 through k in the schedule π are uniquely determined and there exists a unique job J_l to be swapped with the job in position $J_{\pi(k+1)}$ for the schedule to become $(k+1, \delta, \phi)$ -Jackson unless $k = n$. Once again, by proposition 11, the job J_l must appear after the job $J_{\pi(k+1)}$ within the schedule π i.e. $l = \pi(j)$ a unique $j > k+1$ and, therefore, in accordance with the way local mutation operator $\mu_\delta^{\text{local}}$ has been introduced above together with definition 6, it follows that the unique pair of positions to be selected for the successful swap is precisely $k+1$ and j and this happens with probability $\frac{1}{\binom{n}{2}/2} > \frac{1}{n^2}$. At the

same time, once again from the construction of the local mutation operator $\mu_\delta^{\text{local}}$ and definition 6, it follows easily that the only transpositions (swaps) that can take place involve positions beyond $\pi(k)$ so that if the schedule does not become $(k+1, \delta, \phi)$ -Jackson it remains (k, δ, ϕ) -Jackson at least. \square

The following lemma gives us an important and simple polynomial time bound on the runtime complexity of the local mutation operator to reach a (δ, ϕ) -Jackson schedule provided no other mutation takes place.

Lemma 14. *Suppose a given initial sequence $\{J_i\}_{i=1}^n$ of jobs is generic (see definition 12). Choose a subset $I \subseteq \{1, 2, \dots, n\}$ of indices with $|I| = |B_\delta|$ and a bijection $\phi : I \rightarrow B_\delta$. Consider the Markov chain \mathcal{M} on the state space $\mathcal{X}(\phi)$ (recall remark 7 and definition 8) determined by the Markov transition matrix $\{p_{\pi \rightarrow \sigma}\}_\pi$ and $\sigma \in \mathcal{X}(\phi)$ where $p_{\pi \rightarrow \sigma}$ is the probability that the permutation (schedule) σ is obtained from the schedule π via an application of the local mutation operator as described above. Then \mathcal{M} is an absorbing Markov chain with the unique absorbing state $\tilde{\pi}(\phi)$ being the (δ, ϕ) -Jackson schedule. For a schedule $\pi \in \mathcal{X}(\phi)$ let T_π denote the random variable measuring the time to absorption when the chain starts at the state π . Then $\forall \pi \in \mathcal{X}(\phi)$ we have $\Pr(T_\pi > n^4) < ne^{-\Omega(n)}$.³*

Proof. Notice that $\mathcal{X}(\phi) = \bigcup_{i=0}^n \mathcal{X}(\phi)_k$ where

$$\mathcal{X}(\phi)_k = \{\pi \mid \pi \text{ is } (k, \delta, \phi)\text{-Jackson}\}.$$

From the construction of the local mutation operator and proposition 11 it is clear that for $\pi \in \mathcal{X}(\phi)_k \exists!$ $\sigma \in \mathcal{X}(\phi)_{k+1}$ such that $p_{\pi \rightarrow \sigma} = \frac{1}{n^2}$ while $p_{\pi \rightarrow \tau} = 0$ whenever $\tau \neq \sigma$. Thereby, the waiting time random variable

$$T_\pi = \sum_{i=k}^{n-1} T_{\pi_i \rightarrow \pi_{i+1}} \quad (10)$$

where $\pi = \pi_k, \pi_{k+1}, \dots, \pi_{n-1}, \pi_n = \tilde{\pi}(\phi)$ is the unique path to reach the (δ, ϕ) -Jackson schedule $\tilde{\pi}(\phi)$ starting with the schedule π_k via consecutive local mutations while $T_{\pi_k \rightarrow \pi_{k+1}}$ denotes the waiting time between the corresponding successive states. Notice that the random variables $T_{\pi_k \rightarrow \pi_{k+1}}$ are geometric and each is bounded above by a geometrically distributed random variable G having success probability $\frac{1}{n^2}$. Thus $\forall k$ with $0 \leq k \leq n-1$ the probability

$$\Pr(T_{\pi_k \rightarrow \pi_{k+1}} > n^3) \leq \left(1 - \frac{1}{n^2}\right)^{n^3} = e^{-\Omega(n)} \quad (11)$$

Since there are totally at most n summands in decomposition 10 we obtain

$$\Pr(T_\pi > n^4) \leq \sum_{k=0}^{n-1} \Pr(T_{\pi_k \rightarrow \pi_{k+1}} > n^3) \stackrel{\text{by 11}}{\leq} n \cdot e^{-\Omega(n)}$$

which is precisely the desired conclusion. \square

We now proceed to define the global mutation operator in detail:

Global Mutation Operator μ^{global} : Given a schedule (i.e. a permutation) π on $(1, 2, \dots, n)$ select a permutation σ on $\{1, 2, \dots, n\}$ uniformly at random and set $\pi := \sigma$.

In words, the global mutation operator simply reorders all the jobs within a given schedule uniformly at random. The property of global mutation that is behind the design of our hybrid 2 + 2-EA is the following:

Lemma 15. *Suppose a given initial sequence $\{J_i\}_{i=1}^n$ of jobs is generic (see definition 12). Let $\epsilon > 0$ be given so that $\delta = P \cdot \epsilon$. Consider the following simple algorithm:*

Start with a permutation (schedule) π on $\{1, 2, \dots, n\}$.

Repeat

³As a matter of fact, thanks to theorem 2, namely, the property in 4, the constant in the big Ω notation in the exponent can be made arbitrarily close to 1 as long as it is smaller than 1.

{ With probability $\frac{1}{n^5}$ apply a global mutation operator to the schedule π as above. Otherwise, apply a local mutation operator (a local mutation operator is applied with probability $1 - \frac{1}{n^5}$). }

Let $\vec{\pi} = (\pi_0, \pi_1, \dots, \pi_l, \dots)$ denote a typical sequence of outputs of the algorithm above (i.e. an element of the probability space for the induced stochastic (in fact, Markovian) process). Let π_t denote a schedule after t repetitions of the algorithm. Consider the following events. Let l and m denote random times (i.e. \mathbb{N} -valued random variables) and let

$$N(l, m) = \{\vec{\pi} \mid l \leq j < i \leq m \implies \pi_j \sim \pi_i\}. \quad (12)$$

In words, $N(l, m)$ is the event that no global mutation takes place between times l and m (recall remark 7, definition 8 and lemma 14). Then, as long $l \leq m \leq l + n^4$ almost surely, we have $\Pr(N(l, m)) \geq 1 - \frac{1}{\Omega(n)}$.

In addition to the random times l and m as above, the next family of events we consider depends on the choices of $I \subseteq \{1, 2, \dots, n\}$ with $|I| = |B_\delta|$ and $\phi : I \rightarrow B_\delta$.

$$Y(l, m, \phi) = \{\vec{\pi} \mid \exists j \text{ s. t. } l \leq j \leq m \text{ and } \pi_j \in \mathcal{X}(\phi)\}. \quad (13)$$

In words, $Y(l, m, \phi)$ is the event that a (k, δ, ϕ) -Jackson schedule has been encountered at least once between the times l and m (see remark 7 and definition 8). We claim that whenever $m \geq l + n^{\frac{1}{\epsilon}+6}$ we have $\Pr(Y(l, m, \phi)) \geq 1 - e^{-\Omega(n)}$.⁴

Proof. Notice that $\Pr(N(l, m))$ is the probability that no global mutation takes place during the consecutive $m - l$ steps. As long as these consecutive “no occurrence” events happen independently with probabilities $1 - \frac{1}{n^5}$ during at most n^4 consecutive time steps it follows that

$$\begin{aligned} \Pr(N(l, m)) &\geq \left(1 - \frac{1}{n^5}\right)^{n^4} = \left(\left(1 - \frac{1}{n^5}\right)^{n^5}\right)^{\frac{1}{n}} \geq \\ &\geq e^{-\frac{1}{\Omega(n)}} \geq 1 - \frac{1}{\Omega(n)} \end{aligned}$$

as claimed.

Since there are totally less than $n^{\frac{1}{\epsilon}}$ choices of the possible bijections ϕ and the global mutations take place independently with probability $\frac{1}{n^5}$ selecting each such ϕ uniformly at random, it follows that the probability of obtaining a (k, δ, ϕ) -Jackson schedule after any single time step is at least $\frac{1}{n^{\frac{1}{\epsilon}+5}}$. Therefore, the probability of not encountering a (k, δ, ϕ) -Jackson schedule after any single time step is at most $1 - \frac{1}{n^{\frac{1}{\epsilon}+5}}$. Once again, by independence together with the assumption that $m \geq l + n^{\frac{1}{\epsilon}+6}$ we deduce that

$$\Pr(\overline{Y(l, m, \phi)}) \leq \left(\left(1 - \frac{1}{n^{\frac{1}{\epsilon}+5}}\right)^{n^{\frac{1}{\epsilon}+5}}\right)^n = e^{-\Omega(n)}$$

where $\overline{Y(l, m, \phi)}$ denotes the event that no (k, δ, ϕ) -Jackson schedule is ever encountered between l^{th} and m^{th} time steps; in other words, the complement of the event $Y(l, m, \phi)$. The desired conclusion that

$$\Pr(Y(l, m, \phi)) = 1 - \Pr(\overline{Y(l, m, \phi)}) \geq 1 - e^{-\Omega(n)}$$

now follows and finishes the proof. \square

The algorithm in the statement of lemma 15 motivates the construction of a hybrid 2 + 2 EA to minimize the maximum lateness of a schedule up to an approximation ratio of $(1 + \epsilon)$ within time polynomial in n (the number of jobs in the schedule) and $\frac{1}{\epsilon}$ with an overwhelmingly high probability. Essentially, the first individual in the population undergoes consecutive mutations exactly as in the algorithm described in the statement of lemma 15 while the second individual stores the fittest schedule (i.e. the schedule having the minimal maximal lateness as in definition 4) encountered up to the current time.

⁴Just as in lemma 14 with the exception that the dual of the property expressed in 4, namely the one expressed in 5 is used instead, the constant in the big Ω notation in both inequalities can be made arbitrarily close to 1 as long as it is smaller than 1.

Hybrid 2 + 2-EA: The initial population $P_0 = (\pi(0), \sigma(0))$ where $\pi(0)$ and $\sigma(0)$ are arbitrary schedules (i.e. permutations on $\{1, 2, \dots, n\}$). At every iteration of the algorithm the following takes place: given a population $P_t = (\pi(t), \sigma(t))$ after t time steps of the algorithm, let $\pi(t+1)$ be obtained from $\pi(t)$ via an application of the global mutation operator with probability $\frac{1}{n^5}$ and via an application of the local mutation operator with a significantly higher complementary probability of $1 - \frac{1}{n^5}$. **IF** $L_{\pi(t+1)} < L_{\sigma(t)}$ **THEN** set $\sigma(t+1) := \pi(t+1)$ **ELSE** set $\sigma(t+1) := \sigma(t)$. Now obtain the new generation after $t+1$ time steps, $P_{t+1} = (\pi(t+1), \sigma(t+1))$.

Theorem 16. *Suppose we start with a generic (see definition 12 and discussion that follows) sequence of jobs $\{J_i\}_{i=1}^n$ and the Hybrid 2 + 2-EA described above runs for at least $n^{\frac{1}{\epsilon}+6} + n^4$ time steps. Then the probability that a schedule having the maximal lateness smaller than $(1+\epsilon)J^*$ has not been encountered is at most $\frac{1}{\Omega(n)}$ where the constant in the Ω notation can be made arbitrarily close to 1. More generally, the probability that a schedule having the maximal lateness smaller than $(1+\epsilon)J^*$ has not been encountered after $\lambda \left(n^{\frac{1}{\epsilon}+6} + n^4\right)$ time steps is at most $\Omega(n)^{-\lambda}$. In particular, the probability that such a schedule is not found after running the hybrid 2 + 2-EA at least $n^{\frac{1}{\epsilon}+7} + n^5$ time steps is $\Omega(n)^{-n}$ which is overwhelmingly small.*

Proof. Recall from theorem 9 that one can always choose a suitable subset of indices $I \subseteq \{1, 2, \dots, n\}$ and a bijection $\phi^* : I \rightarrow B_\delta$ such that any (δ, ϕ^*) -Jackson schedule π^* achieves the maximal lateness no bigger than $(1+\epsilon)J^*$. Under the mild assumption that the given sequence of jobs $\{J_i\}_{i=1}^n$ is generic (see proposition 11, definition 12 and the discussion following definition 12) such a (δ, ϕ^*) -Jackson schedule is unique. According to lemma 15 a (k, δ, ϕ^*) -Jackson schedule will be encountered at least once as the first individual of the population between times 0 and $n^{\frac{1}{\epsilon}+6}$ is at least $1 - e^{-\Omega(n)}$. In other words, the probability of the event

$$\Pr \left(Y(0, n^{\frac{1}{\epsilon}+6}, \phi^*) \right) \geq 1 - e^{-\Omega(n)}. \quad (14)$$

Now let l_{first} denote the first (random) time when a (k, δ, ϕ^*) -Jackson schedule has been encountered and consider the event $N(l_{\text{first}}, l_{\text{first}} + n^4)$ as in lemma 15. Recall from lemma 15 that $N(l_{\text{first}}, l_{\text{first}} + n^4)$ is the event that no global mutation takes place between the times l_{first} and $l_{\text{first}} + n^4$ and the fact that

$$\Pr \left(N(l_{\text{first}}, l_{\text{first}} + n^4) \right) \geq 1 - \frac{1}{\Omega(n)} \quad (15)$$

Since our goal is to encounter the (δ, ϕ^*) -Jackson schedule, we also need to consider the random time r_{first} when the (δ, ϕ^*) -Jackson schedule has been encountered for the first time after the first (k, δ, ϕ^*) -Jackson schedule has been encountered as well as the events

$$E = \{\vec{\pi} \mid r_{\text{first}} \leq l_{\text{first}} + n^4\} \text{ and } E \cap N(l_{\text{first}}, l_{\text{first}} + n^4). \quad (16)$$

Given that the event $N(l_{\text{first}}, l_{\text{first}} + n^4)$ has taken place, according to lemma 14, the conditional probability

$$\Pr \left(E \mid N(l_{\text{first}}, l_{\text{first}} + n^4) \right) \geq 1 - n \cdot e^{-\Omega(n)} \quad (17)$$

so that

$$\begin{aligned} & \Pr \left(E \cap N(l_{\text{first}}, l_{\text{first}} + n^4) \right) = \\ &= \Pr \left(E \mid N(l_{\text{first}}, l_{\text{first}} + n^4) \right) \cdot \Pr \left(N(l_{\text{first}}, l_{\text{first}} + n^4) \right) \geq \\ & \stackrel{\text{thanks to inequalities 17 and 15}}{\geq} \left(1 - n \cdot e^{-\Omega(n)} \right) \cdot \left(1 - \frac{1}{\Omega(n)} \right) \geq \\ & \geq 1 - \frac{1}{\Omega(n)} \end{aligned} \quad (18)$$

In words, the event

$$\text{LocalGoal}_{\text{first}} = E \cap N(l_{\text{first}}, l_{\text{first}} + n^4) \quad (19)$$

is the event that no global mutation takes place between the times l_{first} and $l_{\text{first}} + n^4$ and, at the same time, the (δ, ϕ^*) -Jackson schedule has been reached between the times l_{first} and $l_{\text{first}} + n^4$ (using local mutation operators only). We summarize inequality 18 as

$$\Pr(\text{LocalGoal}_{\text{first}}) \geq 1 - \frac{1}{\Omega(n)}. \quad (20)$$

Notice the important fact that the events $\text{LocalGoal}_{\text{first}}$ and $Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)$ are independent: indeed, according to the strong Markov property,

$$\begin{aligned} \Pr(\text{LocalGoal}_{\text{first}} | Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)) &= \\ &= \Pr(\text{LocalGoal}_{\text{first}} | \overline{Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)}) \end{aligned}$$

where $\overline{Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)}$ denotes the complement of the event $Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)$, so that the desired conclusion follows from an elementary property of conditional probability.

Consider now the sub-event

$$Z(0, n^{\frac{1}{\epsilon}+6} \phi^*) = Y(0, n^{\frac{1}{\epsilon}+6} \phi^*) \cap \text{LocalGoal}_{\text{first}}. \quad (21)$$

In words, $Z(0, n^{\frac{1}{\epsilon}+6} \phi^*)$ is the event that a (k, δ, ϕ^*) -Jackson schedule has been encountered some time within the first $n^{\frac{1}{\epsilon}+6}$ time steps and, as soon as it was encountered, no global mutation took place during consecutive n^4 time steps and, during these consecutive n^4 time steps, the (δ, ϕ^*) -Jackson schedule has been encountered. In particular, if the event $Z(0, n^{\frac{1}{\epsilon}+6} \phi^*)$ takes place then the (δ, ϕ^*) -Jackson schedule has been encountered within at most $n^{\frac{1}{\epsilon}+6} + n^4$ time steps so that the event

$$\begin{aligned} \text{Target}(n^{\frac{1}{\epsilon}+6} + n^4) &= \{\bar{\pi} \mid \exists j \text{ with } 0 \leq j \leq n^{\frac{1}{\epsilon}+6} + n^4 \\ &\text{and } \pi_j \text{ is } (\delta, \phi^*)\text{-Jackson}\} \supseteq Z(0, n^{\frac{1}{\epsilon}+6} \phi^*) \end{aligned} \quad (22)$$

Thanks to the set inclusion relation expressed in 22 together with the independence of the events $\text{LocalGoal}_{\text{first}}$ and $Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)$, we finally deduce that

$$\begin{aligned} \Pr(\text{Target}(n^{\frac{1}{\epsilon}+6} + n^4)) &\geq \Pr(Z(0, n^{\frac{1}{\epsilon}+6} \phi^*)) = \\ &= \Pr(Y(0, n^{\frac{1}{\epsilon}+6} \phi^*)) \cdot \Pr(\text{LocalGoal}_{\text{first}}) \geq \\ &\stackrel{\text{from inequalities 14 and 20}}{\geq} \left(1 - e^{-\Omega(n)}\right) \cdot \left(1 - \frac{1}{\Omega(n)}\right) = \\ &= 1 - \frac{1}{\Omega(n)} \end{aligned} \quad (23)$$

Inequality 23 tells us that the (δ, ϕ^*) -Jackson schedule is encountered as the first individual in the population of our hybrid 2 + 2-EA within the first $n^{\frac{1}{\epsilon}+6} + n^4$ time steps with probability at least $1 - \frac{1}{\Omega(n)}$. According to theorem 9 the (δ, ϕ^*) -Jackson schedule provides the desired approximation of the optimal fitness with the approximation ratio of $(1 + \epsilon)$ and the first assertion of theorem 16 is established. To see the next assertion, consider the complementary event $\mathcal{N}(1)$ that the optimal schedule is not encountered during the first $n^{\frac{1}{\epsilon}+6} + n^4$ time steps of our hybrid 2 + 2-EA. Clearly

$$\Pr(\mathcal{N}(1)) \leq 1 - \left(1 - \frac{1}{\Omega(n)}\right) = \frac{1}{\Omega(n)} \quad (24)$$

By the Markov property the probability of the event $\mathcal{N}(1 \rightleftharpoons 2)$ that the (δ, ϕ^*) -Jackson schedule is never encountered after the first $n^{\frac{1}{\epsilon}+6} + n^4$ time steps and before $2(n^{\frac{1}{\epsilon}+6} + n^4)$ time steps is bounded above in the same way as the probability of the event $\mathcal{N}(1)$ and therefore it follows that the probability that the (δ, ϕ^*) -Jackson schedule is never encountered before the first consecutive $2(n^{\frac{1}{\epsilon}+6} + n^4)$ time steps is at most

$Pr(\mathcal{N}(1)) \cdot Pr(\mathcal{N}(1 \rightleftharpoons 2)) \leq \frac{1}{\Omega(n)} \cdot \frac{1}{\Omega(n)} = \left(\frac{1}{\Omega(n)}\right)^2$. Continuing in this manner inductively, we deduce that $\forall \lambda \in \mathbb{N}$ the probability that the (δ, ϕ^*) -Jackson schedule is never encountered after the first $\lambda \left(n^{\frac{1}{\epsilon}+6} + n^4\right)$ time steps is at most $\Omega(n)^{-\lambda}$. In particular, when $\lambda = n$ we deduce that the probability of never encountering the (δ, ϕ^*) -Jackson schedule after $n^{\frac{1}{\epsilon}+7} + n^5$ time steps is at most $\Omega(n)^{-n}$ immediately implying the last conclusion. \square

4 Conclusions and Future Work

In the current paper we have designed a parameterized family of hybrid 2 + 2 EAs for the single machine scheduling problem as described in section 3 (alternatively, see [7] for a more detailed description of the single machine scheduling problem with no precedence constraints). The hybrid EA combines local and global mutation operators. The local mutation drives towards a “partial Jackson rule” where the notion of “partial” depends largely on the approximation parameter ϵ , while the global mutation operator is the pure random search. The main result of this work is theorem 16 where we have shown that for a given approximation parameter ϵ and the corresponding local mutation operator that depends on $\delta = \epsilon \cdot P$ where $P = \sum_{i=1}^n p_i$ is the sum of all the processing times of the given jobs to schedule, after $\lambda \left(n^{\frac{1}{\epsilon}+6} + n^4\right)$ time steps our hybrid EA achieves a solution having fitness $(1 + \epsilon)J^*$ with probability $\Omega(n)^{-\lambda}$ where J^* denotes the optimum fitness and the constant in the Ω notation can be made arbitrarily close to 1 (see definition 1 and inequalities 4 and 5 within theorem 2). Our design and complexity analysis is based on rather elementary tools presented in section 2 together with basic Markov chain theory. Since hybrid evolutionary algorithms are rather popular nowadays and many of them combine local mutations driving towards an implementation of a certain well-known algorithm for a specific optimization problem, and global mutations which serve as some kind of a random search (see [1], [2], [3], [4], [8], [9] and [10] as well as a rather extensive discussion in the introduction) we hope that the mathematical ideas and techniques presented in this article can be successfully applied to analyze many other hybrid evolutionary algorithms.

Acknowledgement:

This work is supported by the EPSRC under Grant EP/I009809/1.

References

- [1] C. Goh, E. Teoh and K. Tan A hybrid evolutionary approach for heterogeneous multiprocessor scheduling. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 13(8):833–846, 2009.
- [2] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. Hyper-heuristics: An emerging direction in modern search technology. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 457–474. Springer New York, 2003.
- [3] E. Hart, P. Ross and D. Corne Evolutionary Scheduling: A Review. *Genetic Programming and Evolvable Machines*, 6(2):191–220, 2005.
- [4] F. Khafa. A hybrid evolutionary heuristic for job scheduling on computational grids. In Ajith Abraham, Crina Grosan, and Hisao Ishibuchi, editors, *Hybrid Evolutionary Algorithms*, volume 75 of *Studies in Computational Intelligence*, pages 269–311. Springer Berlin / Heidelberg, 2007.
- [5] G. Grimmett and D. Stirzaker *Probability and Random Processes*, Oxford University Press, 2001.
- [6] J. Lenstra and A. Rinnooy Kan Complexity of Machine Scheduling Problems. *Annals of Discrete Mathematics*, 1:343–362, 1977.
- [7] L. Hall. Approximation Algorithms for Scheduling. In D. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, pages 1–45. PWS Publishing Company, 1997.

- [8] N.K. Bambha, S.S. Bhattacharyya, J. Teich, and E. Zitzler. Systematic integration of parameterized local search into evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):137–155, 2004.
- [9] P. Galinier and J.K. Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3(4):379–397, 1999.
- [10] P. Preux and E.G. Talbi. Towards hybrid evolutionary algorithms. *International transactions in operational research*, 6(6):557–570, 1999.
- [11] R. Goldberg, *Methods of Real Analysis*, John Wiley & Sons Inc., 1976.